

# Assignment 1 Report

CS6400 Spring '17 Team 73

## **Table of Contents**

|  |           |
|--|-----------|
| <b>SQL Data Types</b>                        | <b>1</b>  |
| <b>Constraints</b>                           | <b>4</b>  |
| <b>Task Decomposition with Abstract Code</b> | <b>5</b>  |
| <b>Appendix</b>                              | <b>16</b> |
| IFD Notes                                    | 16        |
| IFD Tasks                                    | 16        |
| IFD Associations                             | 18        |

# SQL Data Types

Database Schema: *cs6400\_sp17\_team073.sql*

**Table: User**

| Name      | Type    | Size | Constraint |
|-----------|---------|------|------------|
| username  | varchar | 250  | NOT NULL   |
| email     | varchar | 250  | NOT NULL   |
| password  | varchar | 50   | NOT NULL   |
| full_name | varchar | 250  | NOT NULL   |

**Table: Site**

| Name           | Type         | Size | Constraint |
|----------------|--------------|------|------------|
| short_name     | varchar      | 250  | NOT NULL   |
| street_address | varchar      | 250  | NOT NULL   |
| city           | varchar      | 250  | NOT NULL   |
| state          | varchar      | 50   | NOT NULL   |
| zip            | unsigned int |      | NOT NULL   |
| contact_number | varchar      | 50   | NOT NULL   |
|                |              |      |            |

**Table: Food Pantry**

| Name               | Type    | Size | Constraint |
|--------------------|---------|------|------------|
| description_string | varchar | 250  |            |

|                    |         |     |  |
|--------------------|---------|-----|--|
| hours              | varchar | 250 |  |
| conditions_for_use | varchar | 250 |  |

**Table: Shelter**

| Name               | Type         | Size | Constraint |
|--------------------|--------------|------|------------|
| description_string | varchar      | 250  |            |
| hours              | varchar      | 250  |            |
| conditions_for_use | varchar      | 250  |            |
| available_bunks    | unsigned int |      |            |
| available_rooms    | unsigned int |      |            |

**Table: Soup Kitchen**

| Name               | Type         | Size | Constraint |
|--------------------|--------------|------|------------|
| description_string | varchar      | 250  |            |
| hours              | varchar      | 250  |            |
| conditions_for_use | varchar      | 250  |            |
| available_seats    | unsigned int |      |            |

**Table: Room**

| Name        | Type         | Size | Constraint |
|-------------|--------------|------|------------|
| room_number | unsigned int |      |            |

**Table: Bunks**

| Name      | Type | Size | Constraint |
|-----------|------|------|------------|
| bunk_type | enum |      |            |

**Table: Item**

| Name            | Type         | Size | Constraint |
|-----------------|--------------|------|------------|
| item_name       | varchar      | 250  |            |
| number_of_units | unsigned int |      |            |
| storage_type    | enum         |      |            |
| Item_type       | enum         |      |            |
| food_category   | enum         |      |            |
| supply_category | enum         |      |            |
| expiration_date | DATETIME     |      |            |

**Table: Client**

| Name              | Type    | Size | Constraint |
|-------------------|---------|------|------------|
| full_name         | varchar | 250  |            |
| id_description    | varchar | 250  |            |
| head_of_household | boolean |      |            |

**Table: Waitlist**

| Name     | Type         | Size | Constraint |
|----------|--------------|------|------------|
| Position | unsigned int |      | unique     |

**Table: Log**

| Name      | Type     | Size | Constraint |
|-----------|----------|------|------------|
| log_entry | varchar  | 250  |            |
| timestamp | DATETIME |      |            |

**Table: Request**

| Name            | Type         | Size | Constraint |
|-----------------|--------------|------|------------|
| request_status  | enum         |      |            |
| units_requested | unsigned int |      |            |
| units_fulfilled | unsigned int |      |            |

## Constraints

### Users

- Must log into system before being able to access protected system services
- Cannot view or modify room waitlist for a site they don't administer

### Non-Users

- Can access only the web views for available bunks/rooms and meals remaining

### Clients

- Must be checked into bunks or rooms at a shelter by users at that shelter
- Cannot be deleted
- Search for clients may not return more than 5 clients

### Site

- Users cannot remove all services from a site

### Shelters

- Bunks cannot be reserved and have no waitlist

### Items

- Items have three storage types - Dry good, Refrigerated, and Frozen
- Items have two main categories - Food or supplies.
- Items categorized as supplies are classified by the following four categories: Personal hygiene, Clothing, Shelter, other.
- Food items are categorized into the following six categories: Vegetables, nuts/grains/beans, Meat/seafood, Dairy/eggs, Sauce/Condiment/Seasoning, and Juice/Drink.

### Requests

- A request may not be approved for more items than exist in a site
- A request may not be made by a foodbank associated with a site administered by the requesting user
- A user should not be able to request an item from a specific food bank

## Task Decomposition with Abstract Code

### Login

#### Task Decomposition:

**Lock Types:** Read-only on User table

**Enabling Conditions:** None

**Frequency:** Frequent

**Schemas:** Single

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

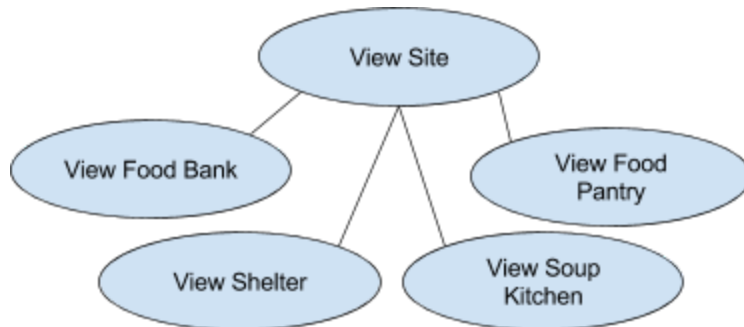
#### Abstract Code:

- User enters *username* (\$Username), *password* (\$Password) input fields.
- If data validation is successful for both *username* and *password* input fields, then when **Enter** button is clicked:
  - If User record is found but User.password != '\$Password':
    - Go back to Login form, with error message.
  - Else:
    - Store login information as session variable \$UserID.

- Go to **View Site** form.
- Else:
  - *username* and *password* input fields are invalid, display **Login** form, with error message.

## **View Site Data**

### **Task Decomposition:**



**Lock Types:** Read only on site table, services tables

**Enabling Conditions:** None, publicly available

**Frequency:** Frequent

**Schemas:** Site table, services table

**Consistency:** Not important

**Subtasks:** Requires mother task, but all tasks can be done in parallel. Display site in top frame, display each service in separate frame

### **Abstract Code:**

- User enters *Short\_Name* (\$short\_name) input fields
- If user administers site:
  - Display attributes of site
  - For each \$service in site:
    - For each \$attribute in \$service
      - Display attribute of \$service
      - If \$attribute is seat\_count, display adjusted seat count
      - If \$service == shelter display bunk count and room count and ***Waitlist*** Button
- Else:
  - Display error
- Go to **View Site Data** form

## **Edit Site Data**

### **Task Decomposition:**

**Lock Types:** Read/write on site/service table

**Enabling Conditions:** Logged in as user who administers site

**Frequency:** Infrequent, but more common to modify service than modify site

**Schemas:** Site table, services tables, attributes of any sub-class

**Consistency:** Important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Display **View Site Data**
- Display dropdown for service type
- Display text field for all *attributes* of any service
- Display **X** button next to each service
- When **Add Service** Button is pressed:
  - Validate input
  - Create new service with displayed attributes
- When **Submit** Button is pressed:
  - Write attributes of site
  - For each \$service:
    - Write attributes of \$service
- When the '**X**' Button is pressed:
  - Delete service entry and any requests linked to service.

## **Create Request**

### **Task Decomposition:**

**Lock Types** Write lock on request table, read on request\_items

**Enabling Conditions:** Logged in as user

**Frequency:** Frequent

**Schemas:** Request table, keys from user and item tables

**Consistency:** Important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Display **View Items** form



- Display the *\$item count* text field next to each item
- When **Submit** button is pressed:
  - Validate input
  - If \$request is for food bank at \$users site:
    - Error
  - Else:
    - Create request of selected item, from \$user, with given count

## **Edit Request**

### **Task Decomposition:**

**Lock Types:** Read/Write lock on request table

**Enabling Conditions:** Logged in as user

**Frequency:** Frequent

**Schemas:** Request table

**Consistency:** Important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Display **View Requests** form
- When the **Submit** button is pressed:
  - Validate input
  - If \$user is not original requester
    - Error
  - Else:
    - Set count of displayed request to updated value.

## **Approve Request**

### **Task Decomposition:**

**Lock Types:** Read/write lock on request table, read/write lock on inventory table

**Enabling Conditions:** Logged in as user for food bank

**Frequency:** Frequent

**Schemas:** Requests, items

**Consistency:** Important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Display **View Requests** outstanding Form

- Display *approved amount* attribute next to each request associated with \$user's food bank (if any)
- When **Approve** button is pressed:
  - If item count at food bank is less than or equal to approved amount
    - Reduce item count at food bank by approved amount
    - Set request status to resolved
  - Else:
    - Error

## **View Requests Outstanding**

### **Task Decomposition:**

**Lock Types:** Read on user table, read on sites table, read on requests table, read on inventory table

**Enabling Conditions:** Logged in as user for food bank

**Frequency:** Frequent

**Schemas:** Requests, sites, users, inventory

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Find \$site administered by \$user
- For each \$request associated with \$site
  - Display \$request.user, \$request.requested count
  - Display \$count of corresponding \$item at food bank (0 if there is no matching item)

Note: Site must have a food bank according to enabling conditions to display requests associated with a given food bank.

## **View Request Status**

### **Task Decomposition:**

**Lock Types:** Read on user table, read on requests table

**Enabling Conditions:** Logged in as user

**Frequency:** Frequent

**Schemas:** Requests, users

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

**Abstract Code:**

- Open all requests associated with \$user
- For each \$request ordered by status, date ascending
  - Display \$request.date, \$request.site, \$request.requested\_count, \$request.approved count, \$request.status

## **Search Clients**

**Task Decomposition:**

**Lock Types:** Read on clients

**Enabling Conditions:** Logged in as user

**Frequency:** Frequent

**Schemas:** Single

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

**Abstract Code:**

- Display text box \$input
- On **Search** button press:
  - Validate the search text box
  - \$client\_list= Find all clients with name like %\$input%
  - If count(\$client\_list)>5, return nothing
  - Else
    - For each \$client
      - Display \$client.id, \$client.name, \$client.head of household status

## **View Client Data**

**Task Decomposition:**

**Lock Types:** Read on clients

**Enabling Conditions:** Logged in as user

**Frequency:** Frequent

**Schemas:** Single

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

**Abstract Code:**

- Display text box \$input
- On **View Client** button press:

- Validate input
- Find client with identifier \$input
- Display \$client.name, \$client.id, and \$client.head\_of\_household

## **View Waitlist**

### **Task Decomposition:**

**Lock Types:** Read on waitlist, read on user, read on site, read on client

**Enabling Conditions:** Logged in as user with shelter

**Frequency:** Frequent

**Schemas:** Single

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Determine shelter associated with site associated with \$user
- For each \$client on waitlist, sorted by waitlist position
  - Display \$client

## **Delete Waitlist Item**

### **Task Decomposition:**

**Lock Types:** Read/write on waitlist

**Enabling Conditions:** Logged in as user associated with shelter

**Frequency:** Infrequent

**Schemas:** Multiple

**Consistency:** Important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Display **Waitlist** form
- Display **X** button next to each waitlist item
- On '**X**' button press:
  - Set \$previous\_position to \$selected\_item.position
  - Remove \$selected\_item
  - For each \$waitlist\_item with position above \$selected\_item.position, decrement \$selected\_item.position

## **Move Waitlist Item**

### Task Decomposition:

**Lock Types:** Read/write on waitlist

**Enabling Conditions:** Logged in as user associated with shelter

**Frequency:** Infrequent

**Schemas:** Multiple

**Consistency:** Important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### Abstract Code:

- Display **Waitlist** Form
- Display text box1 (\$selected\_item)
- Display text box1 (\$New\_position)
- On '**(Up Arrow)**' or '**(Down Arrow)**' button press (up arrow and down arrow are represented as arrow icons):
  - Validate input
  - Set \$previous\_position to \$selected\_item.position
  - Set \$selected\_item.position to \$new\_position
  - For each \$waitlist\_item with position above \$previous\_position:
    - Decrement \$selected\_item.position
  - For each \$waitlist\_item with position below \$selected\_item.position:
    - Increment \$selected\_item.position

## Search Inventory Items

### Task Decomposition:

**Lock Types:** Read on inventory, read on site, read on user

**Enabling Conditions:** Logged in as user

**Frequency:** Frequent

**Schemas:** Multiple

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### Abstract Code:

- Display a text box for \$expiration\_date (default '')
- Display a dropdown for \$storage\_type (containing all known storage types)
- Display a dropdown for \$type (containing 'food' and 'supply')
- Display a dropdown for \$category (containing initially all categories of food and supply)
- Display a text box for \$keyword (default '')
- On dropdown select:
  - Select all inventory matching new restrictions.
  - For each of: \$storage\_type, \$type, \$category:

- Revise dropdown to only include categories that have inventory matching constraints
  - Display \$inventory matching constraints
- On **Search** button press:
  - Select all inventory matching new restrictions.
  - For each of: \$storage\_type, \$type, \$category:
    - Revise dropdown to only include categories that have inventory matching constraints
  - Display \$inventory matching constraints

## **Change Inventory Count**

### **Task Decomposition:**

**Lock Types:** Read/write on inventory, read on users, read on site

**Enabling Conditions:** Logged in as user associated with food bank

**Frequency:** Moderate

**Schemas:** Multiple

**Consistency:** Important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Display **Search Inventory** item form.
- Display text field \$count next to each found item
- On **Search** button press:
  - For \$item with non-empty \$field value
    - If \$item not in \$users food bank
      - Display error, continue
    - Else:
      - \$item.count=\$count

## **Add Inventory Item**

### **Task Decomposition:**

**Lock Types:** Write on inventory, read on user, read on site

**Enabling Conditions:** Logged in as user associated with food bank

**Frequency:** Moderate

**Schemas:** Multiple

**Consistency:** Important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- Display dropdown for \$item\_type (Containing all available item types)

- Display dropdown for *\$item\_category*.
- Display text box for *\$expiration\_date* (default 01/01/9999)
- Display text box for *\$description*.
- On button press:
  - Validate input
  - Create new item with the site associated with the user, attributes according to input fields
  - Reset fields

## **View Meals**

### **Task Decomposition:**

**Lock Types:** Read on items

**Enabling Conditions:** None

**Frequency:** Moderate

**Schemas:** Single

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- $\$v = \text{sum}(\text{count})$  of *\$items* where *\$item.category* = vegetables
- $\$n = \text{sum}(\text{count})$  of *\$items* where *\$item.category* = nuts or *\$item.category* = grains or *\$item.category* = beans
- $\$p = \text{sum}(\text{count})$  of *\$items* where *\$item.category* = meat or *\$item.category* = seafood or *\$item.category* = dairy or *\$item.category* = eggs
- Display "Total meals =" +  $\min(\$v, \$n, \$p)$
- Display "Need more " +  $\text{argmin}(\$v, \$n, \$p)$

## **View Bunks/Rooms**

### **Task Decomposition:**

**Lock Types:** Read on site, rooms

**Enabling Conditions:** None

**Frequency:** Frequent

**Schemas:** Mixed

**Consistency:** Not important

**Subtasks:** Mother Task is not needed. No decomposition needed.

### **Abstract Code:**

- $\$found = \text{false}$
- For each *\$shelter*..:

- \$male=\$shelter.male\_bunks
- \$female=\$shelter.female\_bunks
- \$mixed=\$shelter.mixed\_bunks
- If \$mixed = 0 and \$male = 0 and \$female = 0:
  - Continue
- \$found= true
- Display \$shelter.name
- Display \$shelter.location
- Display \$shelter.phone\_number
- Display \$shelter.hour\_of\_operations
- Display \$shelter.conditions
- Display \$male
- Display \$female
- Display \$mixed
- If not \$found:
  - Display "Sorry, all shelters are currently at maximum capacity"



## Appendix

This section is provided for clarity only and was only included to help clarify the meaning of task names and forms as shown in the Information Flow Diagram.

### IFD Notes

User = ASACS workers

### IFD Tasks

| Task Category | Task Name                  | Short Description   |
|---------------|----------------------------|---|
| Login         | User Login                 | Allows ASACS workers to log into account  |
|               |                            |   |
| Site          | View Site Information      | Allows users to view the information pertaining to a given site   |
|               | Edit Site Information      | Allows users to change information for a given site   |
|               |                            |   |
| Requests      | Create Request             | Allows ASACS workers to create a new request for a given site for food from a food bank. A submitted request will change status to pending.             |
|               | Edit Request               | Change current requests created by user. Either the food bank user or the requestor can change the data for a given request.                            |
|               | View Requests Outstanding  | If the user is associated with a food bank then this will be for the outstanding requests into the specific food bank that the user is associated with. |
|               | View Request Status Report | Allows users to view all outstanding requests into a food bank that the user has created.   |
|               | Approve Request            | Allows users assigned to food banks to approve a  |

|             |                                    |  |
|-------------|------------------------------------|--|
|             |                                    | given set of requests. This would simply be changing the status of the request from pending to approved.   |
|             |                                    |  |
| Web Reports | View Available Bunks/Rooms         | Unauthenticated view that provides a granular list of all available bunks and rooms per shelter.   |
|             | View Meals Remaining               | Unauthenticated view of the current food supply levels at all food banks.  |
|             |                                    |  |
| Client      | Search Clients / View Results      | Provides a search input for the user to find a given client of the ASACS. Allows user to select from top 5 suggested clients based on input values.  |
|             | View Client Report                 | Shows the selected client's data.  |
|             | Edit Client Data                   | Allows users to edit the currently selected client's data. This allows modifying contact info, add service log entry, or place them on waitlist.   |
|             |                                    |  |
| Wait List   | View Wait List Report              | Shows the current shelter room wait list state for the user.   |
|             | Delete wait list entry             | Remove a client from the wait list.  |
|             | Move wait list entry               | Adjust the ordering of clients in the wait list.   |
|             |                                    |  |
| Items       | Search Items / View Search Results | Provides a search input for a user that will search all food banks and provides the search results in a view. This allows users to select an item for either a new or currently not submitted request. |
|             | Add inventory item                 | Allows users to add a specific item into the food bank that they are associated with.  |
|             | Change Item Count                  | Users associated with a specific food bank can change the item count for only that food bank.  |

## IFD Associations

| <b>Task Category</b> | <b>Task Name</b>              | <b>Associated Form</b>            |
|----------------------|-------------------------------|-----------------------------------|
| Login                | User Login                    | User Login Form                   |
|                      |                               |                                   |
| Site                 | View Site Information         | Site Information Form             |
|                      | Edit Site Information         | Site Information Form             |
|                      |                               |                                   |
| Requests             | Create Request                | Edit Request Form                 |
|                      | Edit Request                  | Edit Request Form                 |
|                      | View Requests Outstanding     | View Requests Form                |
|                      | View Request Status Report    | View Requests Form                |
|                      | Approve Requests              | Edit Request Form                 |
|                      |                               |                                   |
|                      |                               |                                   |
| Web Reports          | View Available Bunks/Rooms    | View Available Bunks/Rooms Form   |
|                      | View Meals Remaining          | View Meals Remaining Form         |
|                      |                               |                                   |
| Client               | Search Clients / View Results | Client Search Input and View Form |

|           |                                    |  |
|-----------|------------------------------------|--|
|           | View Client Report                 | Client Report Form                           |
|           | Edit Client Data                   | Client Report Form                           |
|           |                                    |  |
| Wait List | View Wait List Report              | Wait List Form                               |
|           | Delete wait list entry             | Wait List Form                               |
|           | Move wait list entry               | Wait List Form                               |
|           |                                    |  |
| Items     | Search Items / View Search Results | Inventory Item Search Input and Results Form |
|           | Add inventory item                 | Edit Inventory Item Form                     |
|           | Change Item Count                  | Edit Inventory Item Form                     |